

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ, ЧИСЛЕННЫЕ МЕТОДЫ И КОМПЛЕКСЫ ПРОГРАММ

УДК 681.5

А. В. ГАГАРИН, Р. К. ГАЗИЗОВ

ГИБРИДНЫЙ ГЕНЕТИЧЕСКИЙ НЕЙРОСЕТЕВОЙ АЛГОРИТМ В ЗАДАЧЕ ИДЕНТИФИКАЦИИ ПАРАМЕТРОВ ЦИФРОВЫХ МОДЕЛЕЙ

В статье предлагается гибридный генетический нейросетевой алгоритм, в котором радиально-базисная нейронная сеть (РБНС) выполняет функцию аппроксиматора целевой функции (ЦФ). Описывается схема использования НС в составе ГА, схема предварительной обработки данных перед обучением НС и сам алгоритм обучения. Приводятся результаты тестирования разработанного алгоритма на сложных синтетических задачах оптимизации, таких как минимизация функций Расстригина и Розенброка с различным количеством параметров. *Идентификация параметров; оптимизация; генетические алгоритмы; нейронные сети*

ВВЕДЕНИЕ

В работе рассматривается задача идентификации параметров $\mathbf{x} = (x_1, x_2, \dots, x_n) \in D_M \subset \mathbb{R}^n$ динамической математической модели $\hat{\mathbf{y}} = M(t | \mathbf{x})$ объекта исследования (ОИ). В теории идентификации систем [1, 2] эта задача называется *параметрическим оцениванием* модели ОИ.

Предположим, что ОИ характеризуется контролируруемыми выходными сигналами $\mathbf{y}(k) = (y_1(k), y_2(k), \dots, y_m(k))$, которые регистрируются в ходе натурального эксперимента в дискретные моменты времени $t_k, k = 1, 2, \dots$. Им соответствуют выходные сигналы $\hat{\mathbf{y}}(k | \mathbf{x}) = (\hat{y}_1(k | \mathbf{x}), \hat{y}_2(k | \mathbf{x}), \dots, \hat{y}_m(k | \mathbf{x}))$ цифровой модели ОИ, определяемой вектором параметров \mathbf{x} . Невязкой $\varepsilon(k, \mathbf{x})$ называется разность между выходными сигналами ОИ $\mathbf{y}(k)$ и настраиваемой модели $\hat{\mathbf{y}}(k | \mathbf{x})$:

$$\varepsilon(k, \mathbf{x}) = \mathbf{y}(k) - \hat{\mathbf{y}}(k | \mathbf{x}).$$

Пусть имеются данные наблюдения $\mathbf{y}(1), \dots, \mathbf{y}(N)$ за ОИ в N различных моментов времени. Тогда задача параметрического оценивания модели заключается в выборе из всего множества параметров $\mathbf{x} \in D_M$ некоторого оптимального вектора \mathbf{x}^{opt} таким образом, чтобы невязка $\varepsilon(k, \mathbf{x}^{\text{opt}}), k = 1, \dots, N$ была по возможности мала.

Как правило, данная задача сводится к задаче минимизации целевой функции (ЦФ) вида:

$$f(\mathbf{x}) = \sum_{k=1}^N \varepsilon(k, \mathbf{x})^T \cdot \mathbf{W} \cdot \varepsilon(k, \mathbf{x}), \quad (1)$$

где \mathbf{W} – матрица весовых коэффициентов.

Данная функция является мерой близости настраиваемой модели к реальному ОИ и отражает качество идентификации ее параметров. Соответственно,

$$\mathbf{x}^{\text{opt}} = \underset{\mathbf{x} \in D_M}{\operatorname{argmin}} f(\mathbf{x}).$$

На практике ищется минимум $f(\mathbf{x})$ с заданным порогом δ , позволяющим контролировать точность получаемого результата. Иными словами, необходимо найти решение \mathbf{x}^* такое, что:

$$f(\mathbf{x}^*) < f(\mathbf{x}^{\text{opt}}) + \delta. \quad (2)$$

Разные запуски алгоритма оптимизации в зависимости от начальной инициализации и других факторов, могут приводить к нахождению множества D_M^* оптимальных решений $\mathbf{x}_1^*, \mathbf{x}_2^*, \dots$. Выбор одного из них должен осуществляться с помощью дополнительных критериев оценки, помимо (2) (например, путем «физической» интерпретации найденных параметров). Обсуждение этого вопроса выходит за рамки данной работы.

Для нахождения минимума $f(\mathbf{x})$, удовлетворяющего условию (2), обычно применяются итерационные методы оптимизации, генерирующие последовательность приближений $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N_0)}\}$ такую, что $\mathbf{x}^{(N_0)} \in D_M^*$ (то есть $\mathbf{x}^{(N_0)}$ удовлетворяет условию (2)). Значение индекса N_0 показывает количество приближений, требуемых для нахождения оптимума ЦФ с заданным порогом. На практике, в дополнение к критерию (2) часто применяется ограничение

по времени, когда оптимизация останавливается по условию $N_0 = N_0^{\max}$.

В зависимости от сложности модели и величины N , ресурсоемкость вычисления $\hat{y}(k | \mathbf{x})$, $k = 1, \dots, N$ может быть велика. Например, время численного моделирования течения многофазной смеси жидкости и газа в пористой среде, широко используемое в моделях нефтяных месторождений, даже на современных суперкомпьютерах с применением коммерческих симуляторов, таких как Shlumberger ECLIPSE, Roxar Tempest MORE, NGT BOS Core¹ и т. д., может исчисляться сутками. Поэтому стандартные подходы определения параметров таких моделей на основе эволюционных стратегий [7], генетических алгоритмов (ГА) [17, 4], комбинации туннельного и градиентного методов [10], а также других алгоритмов с возможностями глобальной оптимизации зачастую оказываются ресурсоемкими из-за относительно большого количества генерируемых приближений. Область применения алгоритмов градиентного спуска в задачах параметрического оценивания, как правило, ограничивается завершающим этапом уточнения решения, что связано с наличием многих локальных оптимумов у функции $f(\mathbf{x})$. В связи с этим приобретает актуальность задача создания устойчивого алгоритма с возможностями глобальной оптимизации, способного решать поставленную задачу с использованием в среднем меньшего количества приближений по сравнению со стандартными подходами.

1. ГИБРИДНЫЙ ГЕНЕТИЧЕСКИЙ НЕЙРОСЕТЕВОЙ АЛГОРИТМ

1.1. Описание алгоритма

Хорошо известно, что использование стандартного ГА предполагает на каждой итерации для каждого набора параметров (хромосомы в терминах ГА) вычисление значений ЦФ, включающее в себя моделирование ОИ (то есть расчет значений $\hat{y}(k | \mathbf{x})$). Предлагаемый гибридный генетический нейросетевой алгоритм (ГА+НС) является модификацией стандартного ГА, в дальнейшем называемого *главным*, на каждой итерации которого создается нейросетевая аппроксимация $\tilde{f}(\mathbf{x})$ целевой функции $f(\mathbf{x})$, предназначенная для получения прогноза $\tilde{\mathbf{x}}^*$ оптимального решения функции $f(\mathbf{x})$. В большинстве задач параметрического оценива-

ния $\tilde{f}(\mathbf{x})$ будет вычисляться значительно быстрее $f(\mathbf{x})$, поскольку не требует предварительного расчета модельных значений $\hat{y}(k | \mathbf{x})$. Поиск $\tilde{\mathbf{x}}^*$ осуществляется с помощью дополнительного ГА, в дальнейшем называемого *вспомогательным*, для которого в качестве ЦФ используется $\tilde{f}(\mathbf{x})$ (то есть значения, вычисляемые нейронной сетью). Найденный прогноз затем добавляется в популяцию потомков главного ГА на текущей итерации и цикл повторяется. Алгоритм ГА + НС состоит из следующих основных шагов (рис. 1).

1) $v = 1$. Полное множество приближений $A = \emptyset$. Начальная инициализация популяции основного ГА (выполняется случайным образом или на основе предоставляемых пользователем данных). Выходные данные: $\mathbf{x}_1^{(1)}, \dots, \mathbf{x}_{N_p}^{(1)}$, N_p – размер популяции.

2) Вычисление значений ЦФ $f(\mathbf{x}_1^{(v)}), \dots, f(\mathbf{x}_{N_p}^{(v)})$ для всех хромосом популяции основного ГА на текущей итерации v .

3) Проверка критериев остановки оптимизации.

4) Пополнение множества $A = A \cup \{(\mathbf{x}_1^{(v)}, f(\mathbf{x}_1^{(v)})), \dots, (\mathbf{x}_{N_p}^{(v)}, f(\mathbf{x}_{N_p}^{(v)}))\}$, формирование на основе него обучающей выборки $L \subset A$ для НС и обучение сети.

5) Поиск прогноза оптимального решения вспомогательным ГА, для которого ЦФ $\tilde{f}(\mathbf{x})$ вычисляется обученной НС. Выходные данные: прогноз оптимального решения $\tilde{\mathbf{x}}^*$.

6) Создание потомков популяции основного ГА на текущей итерации v . Выходные данные: $\{\mathbf{x}_1^{(v+1)}, \dots, \mathbf{x}_{N_p-1}^{(v+1)}, \tilde{\mathbf{x}}^*\}$ (первые $N_p - 1$ хромосом генерируются с помощью эволюционных операторов главного ГА, а последняя хромосома добавляется с предыдущего шага).

7) $v = v + 1$. Переход к шагу 2.

Адекватный прогноз $\tilde{\mathbf{x}}^*$ оптимального решения может значительно ускорить эволюционный поиск основного ГА. Если прогноз окажется неудачным, эволюционные механизмы обеспечат стабильность оптимизации. Приближения \mathbf{x} , генерируемые основным ГА на протяжении всего процесса оптимизации, вместе с соответствующими значениями ЦФ $f(\mathbf{x})$ участвуют в формировании обучающей выборки для НС (стандартный ГА хранит лишь текущую

¹ Разработка ООО «РН-УфаНИПИнефть»

популяцию фиксированного размера). Вспомогательный ГА может выполнять значительно более интенсивный поиск по сравнению с главным ГА (большой размер популяции, много итераций и т. п.) и с большой вероятностью находить глобальный оптимум \tilde{f} , так как данная ЦФ вычисляется сравнительно быстро.

В известных публикациях, посвященных гибридизации ГА и НС, рассматриваются схемы, предназначенные для условий неповторяемого эксперимента с необходимостью проверки оптимальности решения и единоразовым обучением НС до начала оптимизации на основе заранее подготовленной обучающей выборки [19, 12, 18]. Эти схемы соответствуют одной итерации нейросетевого контура предлагаемого алгоритма ГА+НС.

Наиболее близкий аналог ГА+НС предлагается в работе [11]. Гибридный алгоритм в [11] не содержит вспомогательного ГА, поскольку на вход НС подается значение ЦФ, а на выходе получается прогноз соответствующего вектора искомых параметров. Однако такая схема имеет очевидный недостаток, заключающийся в некорректности обратной задачи, решаемой НС. Одному и тому же значению ЦФ может соответствовать несколько (или даже бесконечно много) различных значений ее аргументов. Это может привести в частности к тому, что ошибка обучения не опустится ниже некоторого значения из-за противоречащих друг другу изменений весов. Тем не менее, авторам [11] удалось добиться существенного увеличения скорости поиска ГА с бинарным кодированием на функциях Растригина и Розенброка, соответственно, от двух и трех аргументов.

1.2. Нейросетевой контур ГА + НС

1.2.1. Выбор архитектуры и алгоритма обучения сети. В алгоритме ГА+НС использовалась радиально-базисная нейронная сеть (РБНС), изначально разработанная для аппроксимации функций многих переменных [2, 9]. К преимуществам РБНС можно отнести необходимость определения количества и параметров нейронов только одного скрытого слоя. Как недостаток можно рассматривать локальность получаемых аппроксимаций [2].

Традиционные алгоритмы обучения РБНС содержат два этапа [14]. На первом этапе определяются позиции центров радиально-базисных функций (РБФ) (т. е. веса первого скрытого слоя), что обычно выполняется с помощью кластеризации обучающей выборки. Здесь и далее предполагается, что в качестве РБФ используется колоколообразная функция Гаусса или по-

добная ей. Для каждой такой функции на первом этапе необходимо также определить параметр δ , задающий ее «ширину» (иногда выделяется как самостоятельный этап). На втором этапе производится настройка весов выходного линейного слоя, что может быть сделано за одну итерацию по методу наименьших квадратов (МНК).

Главными недостатками такой схемы являются необходимость априорного задания количества РБФ (нейронов скрытого слоя) и связанная с ней проблема переобучения РБНС, когда нейронов в скрытом слое берется слишком много. Для преодоления этих недостатков был разработан специальный алгоритм кластеризации, автоматически определяющий не только координаты центров кластеров, но и их количество. Он основан на методе детерминированного отжига (Deterministic Annealing) [16] и использует дополнительную информацию в виде значений ЦФ для каждого входного вектора. Поскольку разработанный алгоритм стремится найти «естественное» количество кластеров в обучающей выборке, в результате создается РБНС с минимальным количеством скрытых нейронов. Принцип минимального размера (parsimonious principle) обеспечивает регуляризацию обучения и позволяет, в частности, получать гладкие аппроксимации. Данное свойство частично решает проблему «проклятия размерности» [5, 2]. По сути, полученные количество и координаты центров кластеров являются эвристически найденным «хорошим» начальным приближением для алгоритма ОРО, запускаемого на втором этапе. Таким образом, на первом этапе закладывается основа для получения хороших обобщающих способностей у обученной сети. Далее приведены этапы алгоритма обучения РБНС в составе нейросетевого контура ГА + НС.

1) Определение количества РБФ и позиций их центров с помощью кластеризации входных данных с автоматическим определением количества кластеров. Параметры σ вычисляются на основе эвристики, предложенной в [2].

2) Уточнение всех весов сети с помощью алгоритма обратного распространения ошибки (ОРО). Такой подход обеспечивает более точную настройку РБНС по сравнению с традиционным расчетом весов только выходного слоя по МНК, так как позволяет корректировать параметры радиально-базисных нейронов скрытого слоя.

3) Возможное дополнительное обучение РБНС на основе алгоритма каскадной корреляции С. Фальмана [8].

1.2.2. Формирование обучающей выборки и области поиска для вспомогательного ГА. На основе кластеризации, проводимой в рамках первого этапа обучения РБНС, формируются обучающая выборка для алгоритма ОРО и область поиска для вспомогательного ГА. Детально данная процедура изображена на рис. 2. Она выполняется на каждой итерации ν главного ГА и включает в себя следующие шаги.

1) Входными данными для дальнейшей обработки является множество $A = \{(\mathbf{x}, f(\mathbf{x}))\}$ всех сгенерированных главным ГА приближений (см. шаг 4 ГА + НС). На итерации ν множество A состоит из $N_A^{(\nu)}$ элементов.

2) Выделение из множества A подмножества из N_B лучших приближений $B \subset A$, где N_B задается пользователем (обычно значения N_B лежат в пределах от 300 до 1000). Целью данного шага является ограничение сверху размера обучающей выборки для РБНС, формируемой в дальнейшем из множества B (множество A растет неограниченно). Обозначим текущее лучшее приближение как \mathbf{x}_b ($\mathbf{x}_b \in B$).

3) Кластеризация B с помощью разработанного алгоритма с автоматическим определением количества кластеров. Результатом является множество из $N_Y^{(\nu)}$ найденных центров класте-

ров $Y = \{\mathbf{y}\}$. Пусть $d(\mathbf{x}, \mathbf{y})$ есть Евклидово расстояние между векторами \mathbf{x} и \mathbf{y} . Обозначим множество точек, принадлежащих j -му кластеру с центром в точке \mathbf{y}_j , как $C_j = \{\mathbf{x}_i \in B : d(\mathbf{x}_i, \mathbf{y}_j) < d(\mathbf{x}_i, \mathbf{y}_k), \forall k \neq j\}$.

4) Формирование обучающей выборки $L \subset B$ для второго этапа обучения РБНС. В выборку L попадают приближения (и соответствующие им значения ЦФ) из N_L ближайших к лучшему приближению \mathbf{x}_b кластеров, то есть

$$L = \{(\mathbf{x}_l, f(\mathbf{x}_l)), \mathbf{x}_l \in \bigcup_{i=1}^{N_L} C_{l_i} : d(\mathbf{x}_b, \mathbf{y}_{l_i}) < d(\mathbf{x}_b, \mathbf{y}_j), \forall j \notin \{l_i\}\}.$$

Значение N_L задается пользователем (в экспериментах данной работы $N_L = 10$).

5) Определение области поиска для вспомогательного ГА, задаваемой в виде двух векторов $\mathbf{a} = (a_1, \dots, a_n)$ и $\mathbf{b} = (b_1, \dots, b_n)$, таких что $a_i = \min\{x_{s,i}\}$, $b_i = \max\{x_{s,i}\}$, $\forall \mathbf{x}_s = (x_{s,1}, x_{s,2}, \dots, x_{s,n}) \in S$. Здесь S — множество приближений из N_S : $N_S \leq N_L$ кластеров вокруг лучшего решения, определяемое аналогично множеству L . То есть область поиска содержится внутри области, покрываемой обучающей выборкой. Значение N_S выбирается пользователем (в экспериментах данной работы $N_S = 4$).

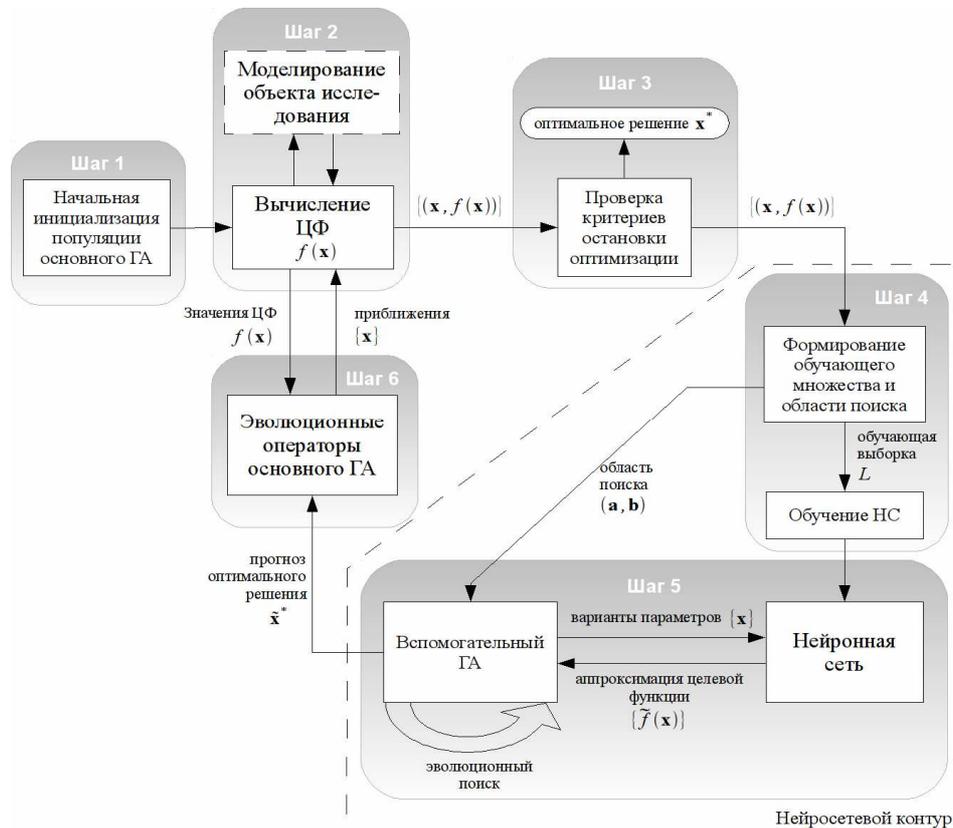


Рис. 1. Схема алгоритма ГА+НС

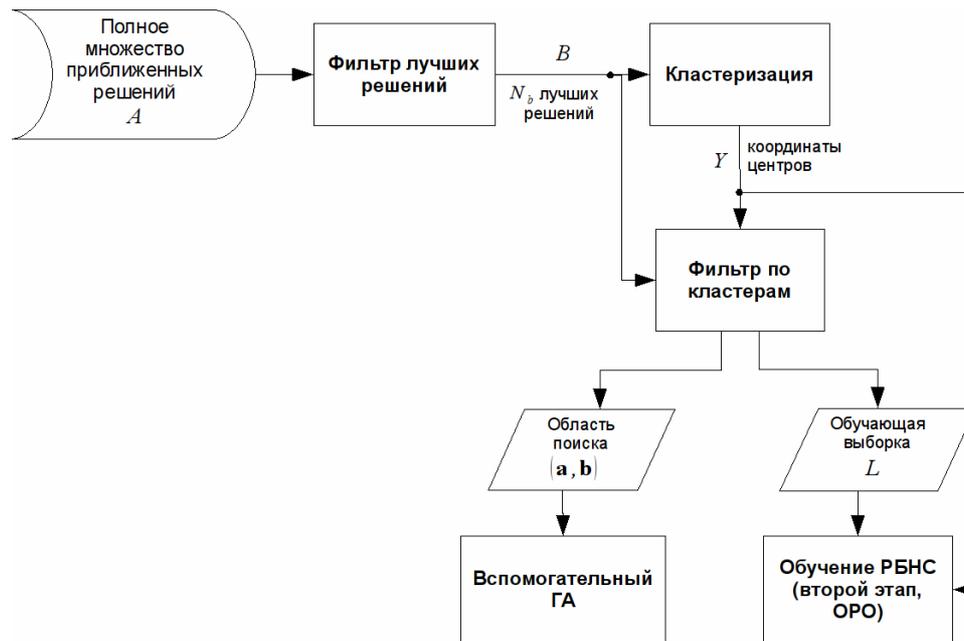


Рис. 2. Схема алгоритма ГА+НС

1.2.3. Дополнительное обучение РБНС.

После второго этапа обучения РБНС с помощью ОРО ошибка обучения может остаться сравнительно большой, а нейросетевая аппроксимация получается очень гладкой. Иногда возникает необходимость уточнить результат, т. е. дополнительно снизить ошибку обучения. Сделать это можно только посредством изменения структуры сети, путем добавления новых скрытых нейронов и перенастройки весов. Для добавления минимально необходимого количества нейронов в данной работе используется идея построения НС по принципу каскадной корреляции (Cascade-Correlation Learning Architecture), предложенная С. Фальманом [8].

Добавляемые скрытые нейроны обучаются по принципу максимизации модуля корреляции с остаточной ошибкой обучения (3):

$$S = \sum_o \left| \sum_p (V_p - \bar{V})(E_{p,o} - \bar{E}_o) \right|, \quad (3)$$

где S – оценка полезности нейрона (больше – лучше); o – индекс нейрона выходного слоя; p – индекс обучающего примера; V_p – выходной сигнал обучаемого скрытого нейрона; $E_{p,o}$ – ошибка нейрона o выходного слоя на обучающем примере p ; \bar{V} , $\bar{E}_{p,o}$ – значения V и E_o , усредненные по всем обучающим примерам.

Для добавления одного нейрона предварительно формируется пул нейронов-кандидатов с радиально-базисными функциями активации, из которых после настройки весов выбирается один, имеющий максимальное значение S . Далее, согласно подходу Фальмана, нейрон

встраивается в РБНС по каскадному принципу, а его веса замораживаются.

Идея обучения РБНС на основе подхода С. Фальмана применялась в [13], где пул нейронов-кандидатов формировался заранее на основе кластеризации по k средним (первый этап обучения) из радиально-базисных нейронов, после чего они добавлялись в скрытый слой. Недостаток метода заключается в используемом алгоритме кластеризации (k -means), требующем априорного задания количества кластеров, то есть максимального количества скрытых нейронов сети.

2. ТЕСТИРОВАНИЕ ГА+НС НА ЗАДАЧЕ МИНИМИЗАЦИИ ТЕСТОВЫХ ФУНКЦИЙ

В данной главе приведены результаты тестирования алгоритма ГА+НС на задаче минимизации функций Растригина и Розенброка [6], которые являются де-факто стандартом проверки алгоритмов оптимизации. Эти функции имеют различный характер и позволяют проанализировать поведение разработанного гибридного нейросетевого алгоритма в сложных условиях (которые с большой вероятностью не могут возникнуть на практике), сравнить его со стандартным ГА и сделать ряд выводов, касающихся его практического применения.

2.1. Описание условий тестирования

Для проверки производительности ГА+НС была проведена серия экспериментов по минимизации функций Растригина и Розенброка от 3, 5, 50 и 100 аргументов. По полученным ре-

зультатам были построены графики изменения лучшего значения ЦФ в процессе ее минимизации алгоритмами ГА+НС, ГА+НС с дополнительным обучением (ГА+НС+ДО) и стандартного ГА. Кроме того, были получены статистические данные, характеризующие поведение алгоритмов в целом. Поскольку ГА имеет стохастическую природу, в целях повышения достоверности результатов для каждой ЦФ и фиксированного числа параметров производилось четыре запуска оптимизации с последующим усреднением результатов. Таким образом, в таблицах и на рисунках далее приведены усредненные величины. Оптимизация останавливалась, если выполнялся хотя бы один из следующих критериев:

- достижение значения ЦФ, меньшего 10^{-7} ;
- достижение максимального количества итераций, равного 100.

Легенда на рисунках расшифровывается следующим образом: GA – ГА; GA+NN – ГА+НС; GA+NN+AL – ГА+НС+ДО (ГА+НС с дополнительным обучением). В таблицах приведены следующие данные: f_b – лучшее найденное значение ЦФ в результате оптимизации; N_o – количество итераций ГА (главного ГА); T – затраченное процессорное время; k_m – количество итераций, на которых нейросетевой прогноз оказался лучшим среди всей популяции (приводится абсолютное значение и процентное, по отношению к общему числу итераций); k_i – коэффициент, показывающий, во сколько раз в среднем удачный нейросетевой прогноз улучшил значение ЦФ. По сути, $k_i = f_{b_{n-1}} / f_b$, где f_b – лучшее значение ЦФ, соответственно, на итерациях $n-1$ и n , причем лучшей особью на n -й итерации является нейросетевой прогноз. Так как решалась задача минимизации, меньшие значения k_i соответствуют лучшему результату.

Настройки основного ГА в алгоритме ГА+НС, а также стандартного ГА в экспериментах были следующими: 60 особей в популяции, 2 элитные особи, ранговая стохастическая равномерная селекция (rank-based stochastic uniform selection), BLX- α кроссовер с вероятностью 0,7, $\alpha = 0,5$; равномерная мутация с вероятностью 0,1; диапазон мутации $[-5,12; 5,12]$ для каждого параметра ЦФ, этот же диапазон использовался для случайной инициализации хромосом первого поколения.

Настройки РБНС: второй этап обучения – модифицированный алгоритм ОРО Resilient Back Propagation (RBP) [15]; максимальное количество итераций алгоритма ОРО – 7000; критерий остановки обучения – падение суммарной

квадратической ошибки (SSE) ниже 0,01; второй критерий остановки – правило «терпения» – если за 100 последовательных итераций SSE изменяется не более чем в 0,001 раза, обучение прекращается. В обучающую выборку отбирались не более 500 точек ($N_B = 500$) из 10 ближайших к лучшему решению кластеров ($N_L = 10$). Область поиска для вспомогательного ГА ограничивалась четырьмя ближайшими к лучшему решению кластерами ($N_S = 4$).

Дополнительное обучение РБНС осуществлялось в соответствии со следующими установками: пул из 8 нейронов-кандидатов, из них на каждой итерации добавлялся только один победитель; максимальное количество итераций алгоритма ОРО для обучения нейронов-кандидатов в пуле – 700; максимальное количество нейронов, добавляемых в процессе дообучения – 100. Выходной линейный слой НС перед началом дообучения подключался, помимо всех скрытых слоев, также напрямую к входному (распределительному) слою.

Перечисленные настройки действовали для обеих ЦФ. Необходимо отметить, что для их определения в данной работе была проведена предварительная серия экспериментов по минимизации функций Растригина и Розенброка с различными параметрами нейросетевого контура и описанными выше критериями остановки.

2.2. Тестирование ГА+НС на функции Растригина

Функция Растригина от n параметров записывается следующим образом:

$$f_{\text{Ras}}(\mathbf{x}) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)), \quad (4)$$

$$\mathbf{x}^{\text{opt}} = (0, \dots, 0).$$

Многоэкстремальность функции Растригина делает градиентные методы неэффективными на задаче ее минимизации. Аппроксимация этой функции также представляет собой сложную задачу. В то же время, при правильно выбранных настройках, стандартный ГА показывает относительно хорошие результаты на функциях, подобных (4).

Результаты экспериментов по минимизации функции Растригина от различного числа параметров алгоритмами ГА, ГА+НС и ГА+НС+ДО представлены в табл. 1–4, а также на рис. 3.

Таблица 1
Статистика оптимизации функции Растригина
от 3 аргументов

Алгоритм	f_b	N_o	T , ч:мин: с	k_m	$k_{m\%}$ %	k_i
ГА+НС	5,57E-008	60,5	21:55:34,43	4,5	7,44	0,6
ГА+НС+ДО	6,86E-008	61,5	38:24:16,25	2	3,25	0,63
ГА	6,05E-008	57,5	00:00:00,03	–	–	–

Таблица 2
Статистика оптимизации функции Растригина
от 5 аргументов

Алгоритм	f_b	N_o	T , ч:мин: с	k_m	$k_{m\%}$ %	k_i
ГА+НС	2,51E-001	92,5	131:30:51,00	2,75	2,97	0,59
ГА+НС+ДО	5,11E-006	98,5	181:57:27,25	1,5	1,52	0,85
ГА	1,01E+000	100	00:00:00,08	–	–	–

Из полученных результатов можно сделать следующие выводы.

1) ГА+НС с дополнительным обучением имеет преимущество во всех экспериментах, вплоть до 100 аргументов функции Растригина.

2) Применение РБНС в экспериментах с 50 и 100 параметрами позволяет скачкообразно уменьшить значение ЦФ в начале оптимизации, за счет чего ее можно остановить гораздо раньше, чем с использованием стандартного ГА. При меньшей размерности входного пространства, ГА+НС и ГА+НС+ДО значительно превосходят стандартный ГА, поскольку нейросетевой контур дает хорошие прогнозы на протяжении всего цикла оптимизации.

3) Относительно большие значения k_i обусловлены усреднением по всем запускам для фиксированного количества аргументов ЦФ (с учетом предыдущего вывода).

4) Поскольку значение функции Растригина вычисляется практически мгновенно, процессорное время, затраченное стандартным ГА, близко к нулю. Процессорное время алгоритмов ГА+НС и ГА+НС+ДО отражает дополнительные вычислительные затраты в нейросетевом контуре. Проведенные эксперименты предназначены для качественной оценки скорости оптимизации сложной ЦФ предлагаемым алгоритмом. Анализ сравнительной ресурсоемкости

реальных задач идентификации параметров моделей ОИ с применением ГА+НС и стандартных подходов выходит за рамки данной статьи.

Таблица 3
Статистика оптимизации функции Растригина
от 50 аргументов

Алгоритм	f_b	N_o	T , ч:мин: с	k_m	$k_{m\%}$ %	k_i
ГА+НС	3,84E+002	100	122:06:56,00	0,25	0,25	0
ГА+НС+ДО	5,11E-006	100	52:13:31,50	2,75	2,75	0,86
ГА	1,01E+000	100	00:00:00,37	–	–	–

Таблица 4
Статистика оптимизации функции Растригина
от 50 аргументов

Алгоритм	f_b	N_o	T , ч:мин: с	k_m	$k_{m\%}$ %	k_i
ГА+НС	1,04E+003	100	103:59:11,75	0,5	0,5	0,78
ГА+НС+ДО	8,37E+002	100	61:47:40,50	3,5	2,5	0,89
ГА	1,11E+003	100	00:00:00,68	–	–	–

2.3. Тестирование ГА+НС на функции Розенброка

Формула функции Розенброка приведена ниже:

$$f_{\text{Ros}}(\mathbf{x}) = \sum_{i=1}^{n-1} \left[100(x_{i+1}^2 - x_i)^2 + (1 - x_i)^2 \right],$$

$$\mathbf{x}^{\text{opt}} = (1, \dots, 1)$$

Эта функция использовалась Де Йонгом для тестирования канонического ГА [6] (впервые предложена Розенброком в 1960 г.). Она представляет собой непрерывную, невыпуклую, унимодальную функцию с минимальным значением, равным нулю, в точке $(1, \dots, 1)$. Минимизация функции Розенброка — очень сложная задача, в том числе и для ГА, поскольку она содержит глубокую параболическую «впадину» вдоль кривой $x_i = x_{i+1}^2$ очень пологой формы [6].

Результаты экспериментов по минимизации функции Розенброка от различного числа параметров алгоритмами ГА, ГА+НС и ГА+НС+ДО представлены в табл. 5–8, а также на рис. 4.

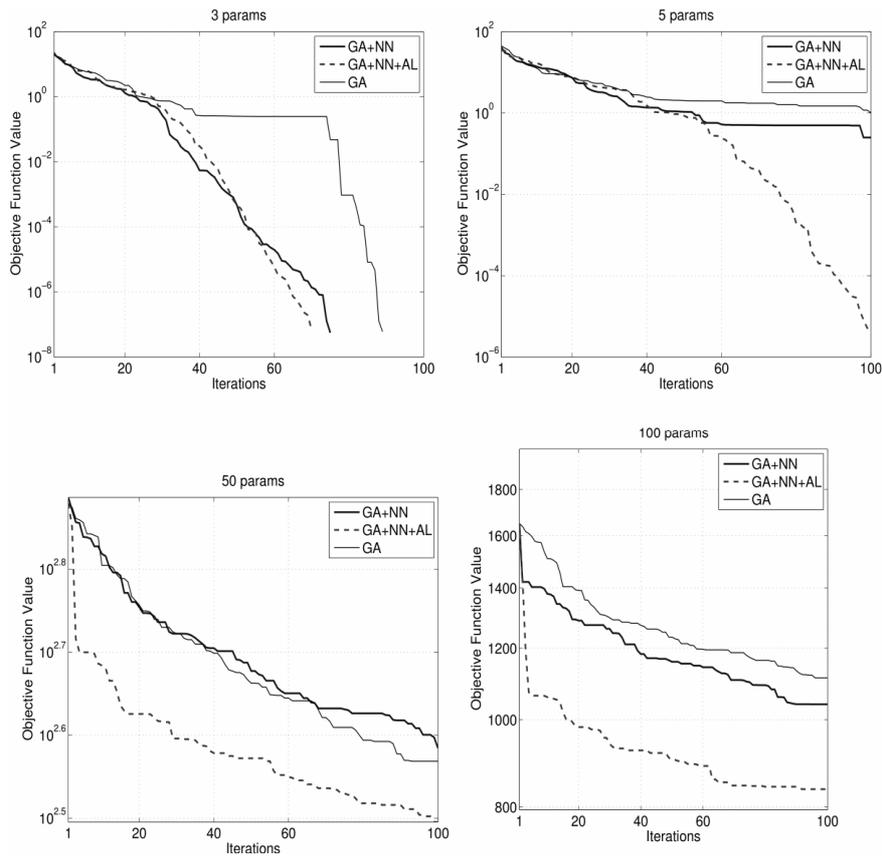


Рис. 3. Сравнение производительности ГА+НС (GA+NN), ГА+НС+ДО (GA+NN+AL) и чистого ГА (GA) на функции Растригина с различным числом параметров

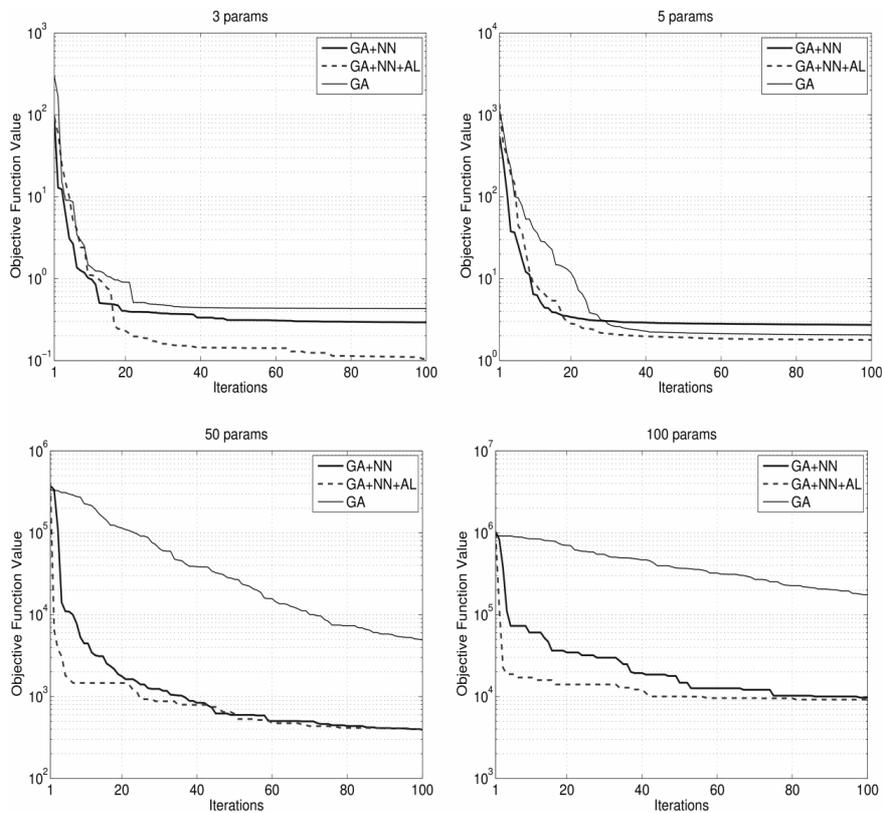


Рис. 4. Сравнение производительности ГА+НС (GA+NN), ГА+НС+ДО (GA+NN+AL) и чистого ГА (GA) на функции Розенброка с различным числом параметров

Таблица 5
Статистика оптимизации функции Розенброка
от 3 аргументов

Алгоритм	f_b	N_o	T , ч:мин:с	k_{nn}	k_{nn} , %	k_i
ГА+НС С	2,93E -001	100	04:23:3 9,75	3	5	0,6
ГА+НС С+ДО	8,37E +002	100	91:08:3 6,75	5	5	0,69
ГА	1,11E +003	100	00:00:0 0,60	-	-	-

Таблица 6
Статистика оптимизации функции Розенброка
от 5 аргументов

Алгоритм	f_b	N_o	T , ч:мин: :с	k_{nn}	k_{nn} , %	k_i
ГА+НС С	2,73E +000	100	03:27: 17,51	6	6	0,61
ГА+НС С+ДО	1,79E +000	100	122:3 4:34,0 0	11, 5	11, 5	0,72
ГА	2,06E +000	100	00:00: 00,08	-	-	-

Таблица 7
Статистика оптимизации функции Розенброка
от 50 аргументов

Алгоритм	f_b	N_o	T , ч:мин:с	k_{nn}	k_{nn} , %	k_i
ГА+НС	3,9 4E+ 002	100	03:37:3 9,14	12, 25	12, 25	0,7
ГА+НС +ДО	4,0 0E+ 002	100	72:53:2 5,25	7	7	0,63
ГА	4,9 2E+ 003	100	00:00:0 0,35	—	—	—

Таблица 8
Статистика оптимизации функции Розенброка
от 50 аргументов

Алгоритм	f_b	N_o	T , ч:мин: :с	k_{nn}	k_{nn} , %	k_i
ГА+НС С	9,72E +003	100	12:01: 43,55	8	8	0,6 3
ГА+НС С+ДО	9,19E +003	100	89:11: 56,25	4	4	0,4 6
ГА	1,75E +005	100	00:00: 00,64	-	-	-

Из полученных результатов можно сделать следующие выводы.

1) На функции Розенброка алгоритмы ГА+НС и ГА+НС+ДО оказались лучшими во всех случаях, за исключением варианта с пятью

аргументами, когда стандартный ГА нашел незначительно лучшее значение ЦФ по сравнению с ГА+НС (рис. 4). ГА+НС+ДО продемонстрировал превосходство во всех случаях без исключения.

2) Преимущество гибридного алгоритма заключается прежде всего в высокой скорости минимизации ЦФ в начале оптимизации. Этот факт становится особенно заметен, когда число аргументов становится большим (50 и 100). В этих случаях ГА+НС и ГА+НС+ДО в течение первых 10–20 итераций резко уменьшают значение ЦФ на 2–3 порядка и серьезно превосходят стандартный ГА.

3) Из табл. 5–8 следует, что процент хороших нейросетевых прогнозов растет с увеличением размерности решаемой задачи, достигая пика примерно к 50 параметрам (12,25% для ГА+НС), после чего начинает постепенно падать. Этот факт подтверждает наличие хороших обобщающих способностей у РБНС, обучаемой по предложенному в работе алгоритму.

ВЫВОДЫ

1. Разработана структурная схема гибридного генетического нейросетевого алгоритма ГА+НС, в котором радиально-базисная нейронная сеть выполняет функцию аппроксиматора ЦФ. Такая схема не зависит от наличия обучающей выборки для НС (наблюдений за ОИ) и достаточно универсальна для решения многих задач оптимизации. Эволюционный поиск главного ГА может быть значительно ускорен хорошим прогнозом лучшего решения, получаемым на выходе нейросетевого контура, в худшем случае эффективность поиска остается на уровне стандартного ГА. Разработанный алгоритм целесообразно применять в задачах с ресурсоемким вычислением ЦФ, например, обусловленным необходимостью в предварительном численном моделировании ОИ (как в задаче параметрического оценивания).

2. Разработан алгоритм предобработки входных данных и обучения РБНС в составе ГА+НС, основанный на кластеризации входных данных с автоматическим определением количества кластеров и координат их центров. По результатам кластеризации строится обучающая выборка и определяется область поиска для вспомогательного ГА.

3. Проведено тестирование разработанного гибридного-нейросетевого алгоритма на сложных задачах минимизации функций Растригина и Розенброка с различным числом аргументов. Результаты тестов показали преимущество ГА+НС практически во всех экспериментах

(особенно с использованием дополнительного обучения). Нейросетевые прогнозы особенно эффективны в начале оптимизации, что хорошо видно в экспериментах с 50 и 100 параметрами функции Розенброка, где отрыв гибридного ГА от стандартного за первые 10 итераций достигает двух–трех порядков значений ЦФ. Подобное поведение ГА+НС дает ему важное преимущество в задачах, когда ЦФ вычисляется на основе результатов предварительных ресурсоемких расчетов, так как позволяет остановить оптимизацию на ранней стадии, получив приемлемое по точности решение.

СПИСОК ЛИТЕРАТУРЫ

1. **Льюнг Л.** Идентификация систем. Теория для пользователя. М.: Наука, 1991. 432 с.
2. **Хайкин С.** Нейронные сети: полный курс. Изд. дом «Вильямс», 2006. 1104 с.
3. **Цыпкин Я. З.** Информационная теория идентификации. М.: Наука. Физматлит, 1995.
4. **Ballester P. J., Carter J. N.** A parallel real-coded genetic algorithm for history matching and its application to a real petroleum reservoir // *Journal of Petroleum Science and Engineering*. 2007. Vol. 59. P. 157–168.
5. **Bellman R.** Adaptive Control Processes: A Guided Tour. Princeton University Press, 1961.
6. **De Jong, K. A.** An analysis of the behavior of a class of genetic adaptive systems: Ph.D. thesis / University of Michigan. — 1975.
7. Evolutionary algorithms applied to history matching of complex reservoirs / R. Schulze-Riegert [et al]. // *SPE Reservoir Evaluation & Engineering*. 2002.
8. **Fahlman S. E.** The recurrent cascade-correlation architecture // *Advances in Neural Information Processing Systems 3*. Morgan Kaufmann, 1991. P. 190–196.
9. **Girosi F., Poggio T., Caprile B.** Extensions of a theory of networks for approximation and learning // *Advances in Neural Information Processing Systems*. Vol. 3. Morgan Kaufmann Publishers, Inc., 1991. P. 750–756.
10. **Gomez S., Poggio T., Caprile B.** Gradient-based history-matching with a global optimization method // *Society of Petroleum Engineering Journal*. 2001. Vol. 6. P. 200–208.
11. **Javadi A. A., Farmani R., Tan T. P.** A hybrid intelligent genetic algorithm // *Advanced Engineering Informatics*. 2005. Vol. 19, № 4. P. 255–262.
12. **Kuo J.-T., Wang Y.-Y., Lung W.-S.** A hybrid neural-genetic algorithm for reservoir water quality management // *Water Res.* 2006. Vol. 40, № 7. P. 1367–1376.
13. **Lehtokangas, M.** Accelerating training of radial basis function networks with cascade-correlation algorithm / M. Lehtokangas, J. Saarinen, K. Kaski // *Neurocomputing*. — 1995. — Vol. 9, no. 2. — Pp. 207–213.
14. **Neruda R., Kudov P.** Learning methods for radial basis function networks // *Future Gener. Comput. Syst.* 2005. Vol. 21, № 7. P. 1131–1142.
15. **Riedmiller M., Braun H.** A direct adaptive method for faster backpropagation learning: the rprop algorithm // *Proc. IEEE International Conference on Neural Networks*. 1993. P. 586–591.
16. **Rose K.** Deterministic annealing for clustering, compression, classification, regression, and related optimization problems // *Proc. of the IEEE*. 1998. Vol. 86, № 11. P. 2210–2239.
17. **Soleng H.** Oil reservoir production forecasting with uncertainty estimation using genetic algorithms // *Proc. Congress on Evolutionary Computation CEC 99*. 1999. Vol. 2.
18. **Srinivas V., Ramanjaneyulu K.** An integrated approach for optimum design of bridge decks using genetic algorithms and artificial neural networks // *Advances in Engineering Software*. 2006. № 38. P. 475–487.
19. **Wang, L.** A hybrid genetic algorithm-neural network strategy for simulation optimization // *Applied Mathematics and Computation*. 2005. Vol. 170. P. 1329–1343.

ОБ АВТОРАХ



Гагарин Александр Владимирович, асп. Дипл. инж. по защите информации (УГАТУ, 2003). Иссл. в обл. алгоритмов искусственного интеллекта, проектирования программн. комплексов.



Газизов Рафаил Кавыевич, проф., зав. каф. высокопроизводит. вычислит. технологий и систем. Дипл. математик (БГУ, 1983). Д-р физ. мат. наук по диф. уравнениям (ИММ Уральск. отд. РАН, 1999). Иссл. в обл. группового анализа диф. уравнений и матем. моделирования.